Heterogeneous industrial vehicle usage predictions: A real case

Heterogeneous Industrial Vehicle Usage Predictions: A Real Case

Dena Markudova
Politecnico di Torino
Turin, Italy
dena.markudova@polito.it

Elena Baralis
Politecnico di Torino
Turin, Italy
elena.baralis@polito.it

Luca Cagliero
Politecnico di Torino
Turin, Italy
luca.cagliero@polito.it

Marco Mellia
Politecnico di Torino
Turin, Italy
marco.mellia@polito.it

Luca Vassio
Politecnico di Torino
Turin, Italy
luca.vassio@polito.it

Elvio Amparore
Tierra Spa
Turin, Italy
eamparore@topcon.com

Riccardo Loti
Tierra Spa
Turin, Italy
rloti@tierratelematics.com

Lucia Salvatori
Tierra Spa
Turin, Italy
lsalvatori@topcon.com

ABSTRACT

Predicting future vehicle usage based on the analysis of CAN bus data is a popular data mining application. Many of the usage indicators, like the utilization hours, are non-stationary time series. To predict their values, recent approaches based on Machine Learning combine multiple data features describing engine status, travels, and roads. While most of the proposed solutions address cars and trucks usage prediction, a smaller body of work has been devoted to industrial and construction vehicles, which are usually characterized by more complex and heterogeneous usage patterns.

This paper describes a real case study performed on a 4-year CAN bus dataset collecting usage data about 2 250 construction vehicles of various types and models. We apply a statistics-based approach to select the most discriminating data features. Separately for each vehicle, we train regression algorithms on historical data enriched with contextual information. The achieved results demonstrate the effectiveness of the proposed solution.

1 INTRODUCTION

Vehicle usage prediction is an established data mining problem, which has found application in both industrial and academic contexts. Since approximately one third of the energy usage is due to transportation [4], predicting vehicle usage is particularly useful for optimizing resources thus reducing the emissions of CO_2 and other pollutant agents. Forecasting key vehicle usage indicators (e.g., utilization hours and fuel consumption levels) is a parallel issue, which is deemed as crucial to optimize many industrial processes [15] such as (i) managing fleets of vehicles in construction sites, (ii) planning periodic maintenance actions on the vehicles of a company, and (iii) optimizing truck routes.

Thanks to the advent of Controlled Area Network (CAN) standards and Internet-of-Things technologies, vehicles are nowadays equipped with smart sensors and tracking systems that capture and transmit high-resolution and multivariate time series data regarding fuel consumption, vehicle movements (e.g., accelerations and drifts), engine conditions (e.g., oil temperature), and route characteristics (e.g., slope). A relevant research effort has been devoted to analyzing CAN bus data by means of supervised Machine Learning techniques [7] to predict the main usage indicator values associated with vehicles. The problem can be modelled as a multivariate time series forecasting task. For example, in [5, 10] the authors applied regression models to predict the future fuel consumption values of trucks based on the past time series values as well as based on the values taken by correlated time series (e.g., travelled distance, average speed, average road slope). The time series describing travel characteristics appeared to be the most discriminating features. Similarly, the studies presented in [2, 13] focused their analyses on CAN Bus and trip data to predict the fuel consumption of cars and trucks. The empirical comparisons reported in the aforesaid studies showed that Support Vector Regression models appeared to be the best performing regression algorithms in the considered scenario. Different types of on-road vehicles were considered in [14], [8], and [3]. They proposed to use Random Forests to learn predictive models for public buses, waste collectors, and heavy duty trucks, respectively. A more extensive review of the literature on on-road vehicle consumption models is given in [15].

Due to the peculiar characteristics of the environment (e.g., construction sites, rural areas) and to their context of use, industrial and construction vehicles show fairly heterogeneous and hardly predictable usage patterns. Hence, a smaller body of work has been devoted to predicting their future usage. Most of the presented works in this field targeted very specific challenges. For example, the works presented in [1, 12] focused on modelling fuel consumption of mining trucks, where a large portion of fuel consumption was due to avoidable idle times. To the best of our knowledge, extensive studies considering and comparing various construction and industrial vehicle types and models with each other have not been presented in literature.

This paper addresses the automatic prediction of the daily utilization hours of industrial and construction vehicles belonging to multiple types and models. It presents a real case study performed on a 4-year real dataset collecting CAN bus data of 2 239 industrial vehicles with various characteristics working in construction sites placed all over the world. The aim of the study is to help site managers to properly schedule short-term fleet management and maintenance actions (e.g., schedule refueling).

To address utilization hours prediction, we apply regression models trained on past vehicle data. Training data consist of CAN bus data (engine rpm, oil temperature, fuel level) enriched with contextual information (e.g., day of the week, season, location). Since the analyzed time series shows non-stationary and rather heterogeneous usage trends (independently of vehicle type and model), we train regression models separately on each vehicle. To tailor prediction models to the most discriminating vehicle characteristics, we apply a statistics-based approach to select the most relevant data features. This reduces the bias due to the presence of uncorrelated variables and thus allows to focus the model training on the most salient information.

The rest of the paper is organized as follows. Section 2 describes the dataset. Section 3 formalizes the problem addressed in the paper and describes the methodology we adopt. Section 4 summarizes the main experimental results, while Section 5 draws conclusions and discusses the future research directions.

2 DATA OVERVIEW

This study focuses on analyzing telematics data acquired from industrial vehicles used in heterogeneous contexts (e.g., construction sites). The data was provided by Tierra S.p.A. Tierra is a company that provides telematics solutions for tracking vehicles of multiple vendors.

Data description. The study encompasses a quite large set of heterogeneous industrial vehicles. Overall, we analyzed data related to 2 239 vehicles belonging to 10 different types and located in 151 different countries spread all over the world. The dataset collects approximately 4-year data (from January 2015 to September 2018). For each vehicle we consider different data characteristics. The main classes of data features are enumerated below.

- *CAN bus* information (e.g., Engine ON/OFF, CAN parametric messages, Diagnostic Messages, and status reports);
- *Vendor* information (e.g., unit/asset info, maintenance services);
- Information provided by *embedded devices* installed onboard (e.g., digital inputs report);
- *Contextual* information
  - *Spatial* information (geographical location of the vehicle, region, country)
  - *Temporal* information (time stamp, day of the week, holiday/working day – depending on the country, week of the year, month of the year, season, year)

CAN messages are generated by the vehicle sensors and the Machine Control Systems at a high frequency (up to 100 Hz) and gathered by a controller, where they are collected and preprocessed. The system then sends an aggregated report to a centralized server every 10 minutes. For each vehicle, the report contains a set of data features describing the engine and vehicle statuses, e.g., fuel level, engine oil pressure, engine coolant temperature, engine fuel rate usage, speed, working hours, percent load, digging press, pump drive temp, oil tank temperature, etc. Based on acquisition time and number of acquired samples we derive the daily utilization hours for each vehicle.

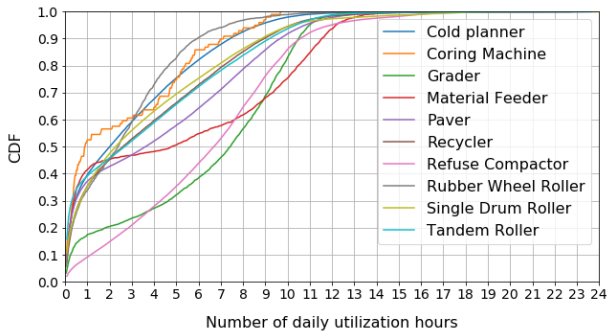
Data preparation. To prepare CAN bus data for the Machine Learning process, we apply the following preparation steps: (i) *Data cleaning*, to handle missing values, make data formats uniform, and verify the absence of data inconsistencies. This is particularly important since vehicles operate in remote regions where the sudden absence of connectivity may affect data collection. (ii) *Normalization*, to normalize the values of continuous features in order to make them comparable with each other, (iii) *Aggregation*, to aggregate feature values on a daily basis. (iv) *Enrichment*, to enrich CAN bus data with multiple-level and multi-faceted contextual information, and (v) *Transformation*, to tailor input data to a relational data format.

Vehicles are characterized by a unique identifier (the *Vehicle id*) and are classified based on the type of construction vehicle (e.g., refuse compactor, single drum roller, tandem roller, coring machine, paver, recycler, cold planner, and grader). Each type is then split into several models (i.e., a type subcategory). For example, the dataset contains vehicles of 44 different models of refuse compactors, 65 models of single drum rollers, 10 models of recyclers, etc. Finally, the dataset contains data for multiple units for each model. CAN bus data has been enriched with contextual information, to allow the exploitation of seasonality and short-term periodic trends to enhance prediction accuracy. For example, for most of the vehicles located in the northern hemisphere, the number of days in which they were unused was maximal in December and January due to Christmas holidays and unfavourable weather.

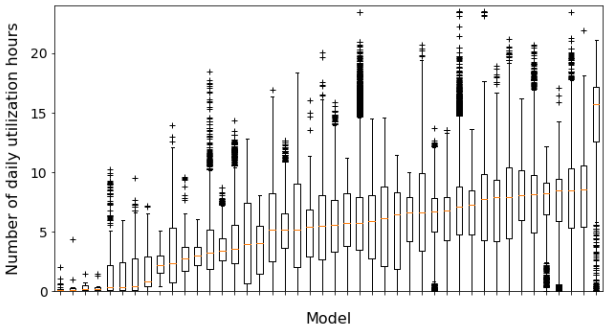
Data characterization. Data characterization is instrumental to discover similarities and differences among vehicle usage patterns. We focus on the number of hours a vehicle is active each day, for the whole considered time period. We remove the days where we did not record any usage. In a Cumulative Distribution Function (CDF), a curve value $F(x)$ indicates the fraction of days where the number of daily utilization hours are less than or equal to x. Figure 1(a) shows the empirical CDF of vehicle usage. The plot highlights the heterogeneity of the vehicle usage distributions across different types. For instance, graders and refuse compactors are used more than 6 hours per day in median, whereas the coring machines showed opposite usage patterns, with a median usage of less than one hour. Some vehicle types expose a long tail in the CDF, meaning that they are sometimes working up to 24 hours per day.

Within specific vehicle types and models, vehicles still show highly variable usage patterns. For example, Figure 1(b) shows the boxplots of the utilization hours for all the 44 models of type refuse compactor (that is the mostly used vehicle type). Models are sorted in ascending order according to their median utilization. The boxplots display the full range of variation (from minimum to maximum), and the first, second and third quartiles. Values with + marker are classified as outliers (deviation of more than 1.5 times interquartile range from the first and third quartiles). The plot confirms a large variance of the utilization hours across vehicle types. Figure 1(c) deepens the analysis into specific vehicle units of the refuse compactor model. For each vehicle, we analyze the utilization hour series across single units. The results indicate that even within the same model, units have very different usage patterns. Finally, Figure 1(d) plots the weekly utilization hours series for five vehicle units at random. Daily patters are even more uncorrelated and noisy. Despite the units being of the same type and model, usage patterns shows non-stationary and uncorrelated trends.

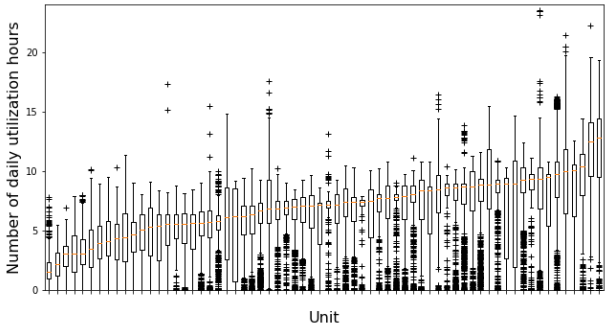
This large variability suggests us to address the prediction problem by training a per-vehicle regression model. Building a model for a vehicle type or model would results in a too generic approach.



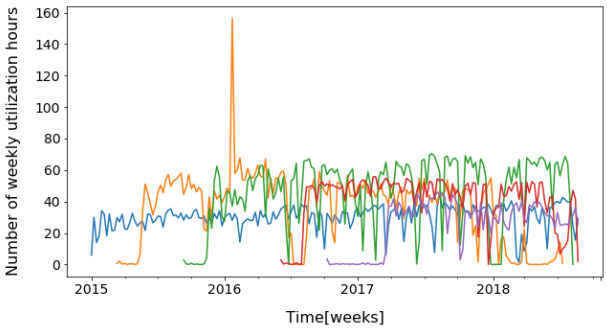
(a) Cumulative Distribution Functions of number of daily utilization hours per vehicle type. Inactive days are removed.



(b) Boxplots of number of daily utilization hours for different models of a single vehicle type (all refuse compactors)



(c) Boxplots of number of daily utilization hours for single units of a specific model of a refuse compactor type.



(d) Time series of weekly utilization hours for 5 different single vehicles of a specific model of refuse compactor.

Figure 1: Data characterization for types, models, and single units. Data is highly heterogeneous.

3 METHODOLOGY

This section formalizes the problem we address in this study. We describe the methodologies to generate per-vehicle training datasets, filter uncorrelated features from training data, and train the regression models.

Problem statement. Let H_t^x be the daily utilization hours of vehicle x on day t. The classical univariate series forecasting problem entails predicting the utilization hours on the next day H_{t+1}^x based on the series of historical values H_t^x, H_{t-1}^x,..., H_{t-w+1}^x within a time window of size w. Since utilization hours are likely to be temporarily correlated with each other, we consider them as a function f of the most recent values within a specified time window TW – hereafter denoted as *training window*. Formally speaking,

$$H_{t+1}^x = f(H_t^x, H_{t-1}^x, \ldots, H_{t-w+1}^x)$$

where H_{t+1}^x is the value of the target variable and f(·) is the prediction function we need to define.

Let $\mathcal{F}$ be the set of CAN bus and contextual features stored in the relational dataset (according to the data description in Section 2), and let F_t^x be the value of an arbitrary feature $F \in \mathcal{F}$ (e.g., engine oil pressure) associated with vehicle x on day t. We can extend the classical forecasting problem to a multivariate context by considering not only the historical values of the utilization hours themselves, but also those of the other features in $\mathcal{F}$. Formally speaking, the problem can be formalized as follows:

$$H_{t+1}^x = f(H_t^x, H_{t-1}^x, \ldots, H_{t-w+1}^x, F_t^x, F_{t-1}^x, \ldots, F_{t-w+1}^x, \ldots)$$

According to our preliminary data exploration, most vehicles are used only for few days a week (e.g., vehicles of type refuse compactor were used 36% of the days in 2017). For this reason, we investigate two variants of the multivariate problem:

- Predict the utilization hours on the next day (*next-day* scenario, in short),
- Predict the utilization hours on the next working day, i.e., the next day on which the vehicle will be used at least 1 hour (*next-working-day* scenario).

Training data generation. Given dataset D, target vehicle x, and training window TW, in this phase we build a per-vehicle training dataset T_x by applying the sliding window approach. More specifically, a smaller window SW slides over the entire training window TW to capture interesting temporal correlations among close utilization hours. Each slide generates a record in the training dataset. For instance, if we set the window size $w = |SW|$ to 7 to capture weekly periodicity, we obtain $|TW| - 7$ training samples, i.e., we can have up to $|TW| - 7$ windows containing the next targeted day and the previous 7 days. In a nutshell, each record contains a target value of utilization hours on arbitrary day $t + 1$, and the feature values associated with each of the previous 7 days, i.e., $t - 6, \ldots, t$.

Statistics-based feature selection. Since the considered vehicles show variable usage patterns, the correlation between the features in the training dataset and the target feature H_{t+1}^x is likely to change from vehicle to vehicle. To tailor prediction models to the most discriminating features, we adopt a statistics-based approach to filter training features prior to regression model

learning. Given a vehicle x, we compute the autocorrelation function [6] of the original daily utilization hour time series to decide which days in the sliding window (among $t-w, \ldots, t$) the past series values) are actually correlated with the target one $(t+1)$. The autocorrelation function estimates the correlation of a series with a delayed copy of itself (adding a time lag). The larger the autocorrelation value associated with an arbitrary lag l, the more correlated the target day $t+1$ with previous day $t+1-l$.

Figure 2 shows an example of autocorrelation function associated with the utilization hours of a specific refuse compactor unit. Obviously, the autocorrelation value is maximal with lag equal to 0, i.e., comparing the series with itself. In the example, the autocorrelation function is able to capture a weekly periodicity (i.e., high autocorrelation value at l=7, 14, 21, ...). Similarly, the day after ($l = 1, 8, 15, \ldots$), and before ($l = 6, 13, \ldots$) also exhibits a high correlation with the target day. Other days instead are marginally correlated.

To exploit this correlation, we pick the K lags $l \in [1, w]$ with maximal autocorrelation value, which correspond to the K days that are mostly correlated with the target day. Then, we select the features in the training dataset corresponding to these days. Formally speaking, let $t^*, t^{**}, \ldots$ be the subset of selected days. The problem is reformulated as follows:

$$H^x_{t+1} = f(H^x_{t^*}, \ldots, H^x_{t^{**}}, \ldots, F^x_{t^*}, F^x_{t^{**}}, \ldots)$$

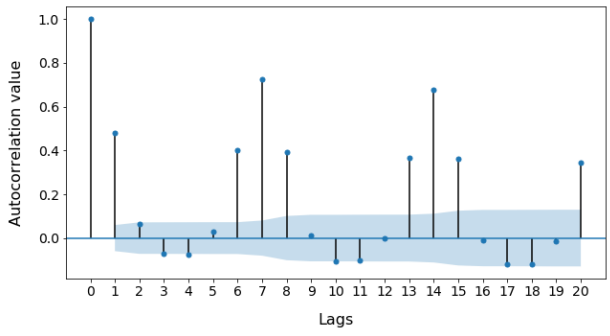


Figure 2: Autocorrelation function for a specific vehicle considering a TW with 20 days.

Regression model learning. To finally build the regression model $f(\cdot)$, we rely on standard approaches. Specifically, we focus on the following algorithms: (i) Linear Regression (LR), (ii) Lasso Regression, (iii) Support Vector Regression (SVR), and (iv) Gradient Boosting (GB). The last approach is an ensemble method (i.e., an ensemble of decision tree models) [7]. To train the models we use the scikit-learn implementations [9]. We also consider two naive baseline methods: (i) Using the last observed value (LV), (ii) using a moving average value (MA) over the past w days.

4 EXPERIMENTAL RESULTS

This section summarizes the most relevant results obtained by applying the explained methodology on the Tierra dataset. We conducted an extensive experimental campaign to (i) analyze the impact on prediction outcome, (ii) test multiple algorithm configurations to identify the best settings in different scenarios, and (iii) estimate the prediction errors to get confidence intervals for the estimations, and (iv) use the best obtained models on vehicles of different models and types. Due to the lack of space, hereafter we will report a selection of the most significant results.

The experiments are performed on an Intel(R) Core(TM) i7-8550U CPU with 16 GB of RAM running Ubuntu 18.04 server.

4.1 Experimental design

To evaluate the performance of the proposed approach we apply a hold-out validation. We adopt two established strategies for time series forecasting, i.e., sliding window and expanding window methods [11]. The sliding window strategy entails the following steps:

(1) Define a fixed-size sliding window SL of up to $w = 150$ days sliding over the whole time period (approximately 4-year data).
(2) For each window slide, prepare the relational dataset using the windowing approach (see *Training data generation* in Section 3) and apply the feature selection step.
(3) Separately for each vehicle train different regression models on the prepared training dataset.
(4) Apply the regression models to predict the utilization hours of the vehicles either on the next day (*Next-day* scenario) or on the next working day (*Next-working-day* scenario).
(5) Evaluate the per-vehicle prediction errors by averaging the errors over the entire period.
(6) Evaluate the overall prediction error by averaging the prediction errors over all the vehicles.

The expanding window strategy differs from the sliding window one because the training window at Step (1) is not fixed-size, but it includes all the preceding days in the original dataset (see Figure 3).

To assess the quality of the prediction outcomes at Steps (5) and (6), we computed the Percentage Error (PE) between predicted and actual utilization hours:

$$PE = 100 \cdot \frac{\sum_{i=1}^{n} |H^i_{pred} - H^i_{actual}|}{\sum_{i=1}^{n} |H^i_{actual}|}$$

4.2 Algorithm settings

For each algorithm we run a grid search to fit the model to the analyzed data distribution. The selected settings are summarized below. More details on the algorithm parameters are provided by [9].

- Lasso: α=0.1
- Support Vector Regressor (SVR): kernel = rbf, C = 10, ϵ = 0.1, γ = 1
- Gradient Boosting (GB): learning rate=0.1, n_estimators = 100, max_depth = 1, loss = lad
- Baseline - Predict the Moving Average (MA): moving average period = 30.

4.3 Effect of the feature selection step

We first analyze the effect of varying the number of considered previous days (K) on the average Prediction Error. Since its effect is influenced by the sliding window size w, we jointly analyzed the two effects. Figure 4 reports results, with one curve per different value of window size w. The results show that

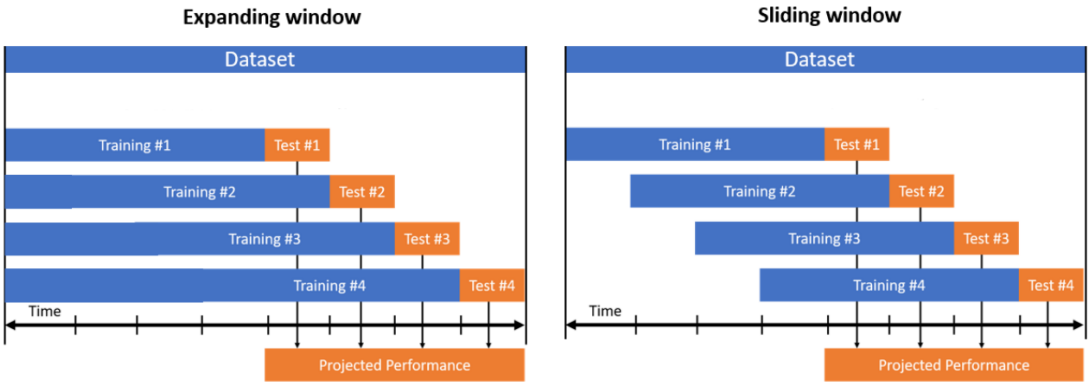


Figure 3: Strategies: expanding window vs. sliding window

- A smart feature selection yields up to 10% improvement in terms of PE.
- The optimal number of considered previous days K ranges between 10 and 30.
- The more previous days you consider, the more features you add in the training dataset.
- Focusing on a limited number of days (< 10) makes prediction models sensitive to noise and data overfitting.
- Including a very large number of features in the training dataset may significantly increase the complexity of the learning generation phase.
- The more training data you get (i.e., larger w), the more robust the model (except for very small K values, for which the robustness of the generated models is not guaranteed).
- Expanding the training window (i.e., increasing $|SW|$ on all past data) performs better, but at the cost of additional computational complexity (not reported - training time grows with data).

To balance model accuracy and complexity, hereafter, we will set $K = 20$ and $w = 140$ (unless otherwise specified).

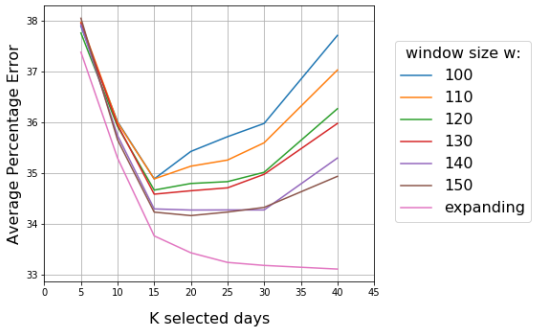


Figure 4: Parameter analysis: effect of the number K of selected days and window widths.

4.4 Analysis of prediction errors

We now compare the performance of different regression models. Figure 5 summarize the results in two considered scenarios (*Next-day* and *Next-working-day*). As expected, Machine Learning approaches perform better than baseline strategies in both cases. Within each strategy, single and ensemble methods achieve similar percentage errors, with SVR performing comparably to Gradient Boosting. Learning an ensemble of multiple models as such does not yield significant performance improvements.

In the *Next-working-day* scenario prediction errors are much better - approaching 15% vs. 30% average error in the *Next-day* scenario. This because removing non-working days simplifies the forecasting problem, being the latter almost randomly present. To exemplify this effect, Figures 6(a) and 6(b) plot the actual and predicted series in the two scenarios. In the *Next-working-day* scenario the predicted curve appears to fit better the actual series thanks to the absence of less-predictable non-working days. Hence, in the contexts in which holidays/non-working days are known in advance, adapting the raw data to this simpler scenario yields significant advantages in terms of model accuracy.

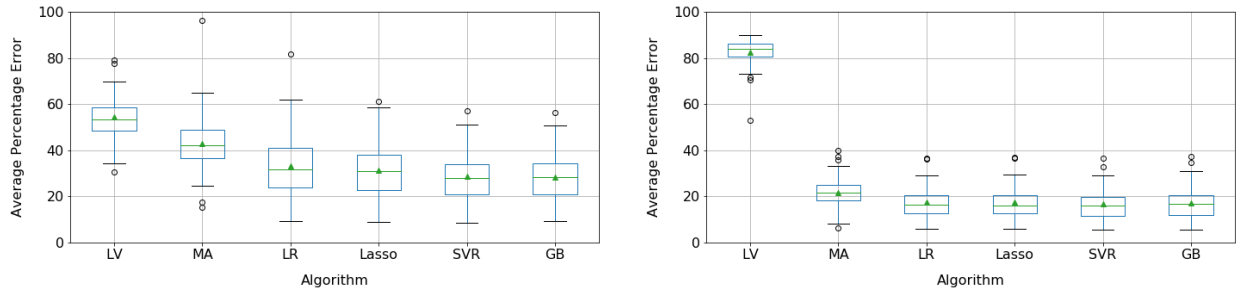
4.5 Prediction time

We measured the execution time of the applied methodology, which includes (i) Data preparation and feature selection, (ii) Model training, and (iii) Model application. Learning the regression models, i.e., Step (ii), turned out to be the most computationally expensive task. According to the performed experiments, the time spent in accomplishing the other tasks is negligible.

The overall execution time taken by single regression models (trained with the settings recommended in Section 4.2) varied between few seconds for simpler models (MA, LV, LR, Lasso) and few tens of seconds for the most complex ones (SVR). The time spent by ensemble methods (i.e., Gradient Boosting) was approximately one order of magnitude higher than single models. However, as discussed in Section 4.4, combining multiple models did not provide significant performance improvements.

5 CONCLUSIONS AND FUTURE WORKS

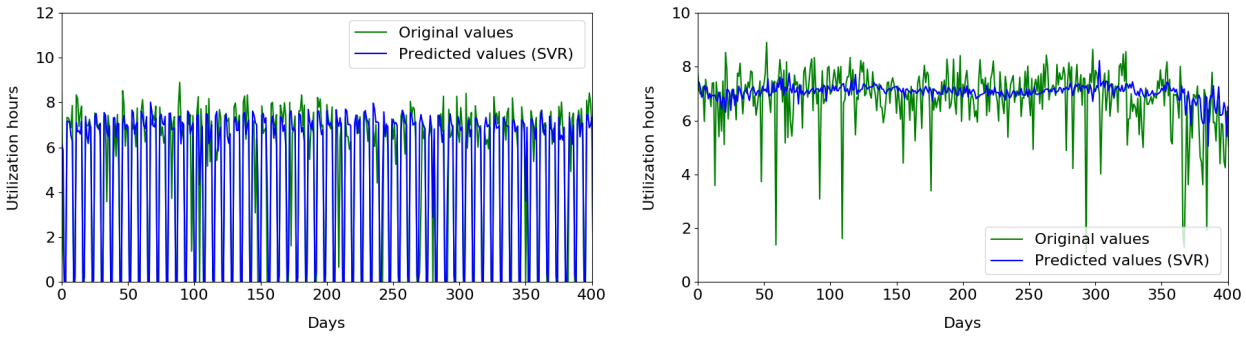
The paper describes our experience in using supervised regression techniques to predict industrial vehicle usage based on the analysis of CAN bus and contextual data. The presented case study corroborates previous studies focused on specific construction vehicle types by considering a broader set of vehicle types and models. To effectively cope with a heterogeneous set of vehicles, we selected ad hoc feature sets tailored to each vehicle prior to model learning. The generated models appeared to be quite effective (i.e., with 15% error) in predicting vehicle utilization hours on the next working day. Without any apriori knowledge about the days of idleness, prediction errors on average double, but for many vehicle types and models it was still possible to



(a) Error distribution. Scenario: *Next-day*.

(b) Error distribution. Scenario: *Next-working-day*.

Figure 5: Algorithm comparison in terms of prediction error



(a) Scenario: *Next-day*: idle days are hard to predict.

(b) Scenario: *Next-working-day*: working hours are easier to predict.

Figure 6: Example of predicted vs. actual values for one unit. Randomness in idle days, and days with limited working time make the prediction hard.

accurately forecast non-stationary trends. Future developments of this research will entail the integration of additional contextual information (e.g., weather) and the use of classification models to predict discrete usage levels.

Acknowledgements

The research leading to these results has been funded by the SmartData@PoliTO center for Big Data and Machine Learning technologies.

REFERENCES

[1] D.M. Bajany, X. Xia, and L. Zhang. 2017. A MILP Model for Truck-shovel Scheduling to Minimize Fuel Consumption. *Energy Procedia* 105 (2017), 2739 – 2745. https://doi.org/10.1016/j.egypro.2017.03.925 8th International Conference on Applied Energy, ICAE2016, 8-11 October 2016, Beijing, China.

[2] Ahmet Gurcan Caapraz, Pinar Ozel, Mehmet Azevkli, and Amer Faruk Beyca. 2016. Fuel Consumption Models Applied to Automobiles Using Real-time Data: A Comparison of Statistical Models. *Procedia Computer Science* 83 (2016), 774 – 781. https://doi.org/10.1016/j.procs.2016.04.166 The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016) / The 6th International Conference on Sustainable Energy Information Technology (SEIT-2016) / Affiliated Workshops.

[3] Oscar F. Delgado, Nigel N. Clark, and Gregory J. Thompson. 2012. Heavy Duty Truck Fuel Consumption Prediction Based on Driving Cycle Properties. *International Journal of Sustainable Transportation* 6, 6 (2012), 338–361. https://doi.org/10.1080/15568318.2011.613978

[4] Kebin He, Hong Huo, Qiang Zhang, Dongquan He, Feng An, Michael Wang, and Michael P. Walsh. 2005. Oil consumption and CO2 emissions in China's road transport: current status, future trends, and policy implications. *Energy Policy* 33, 12 (2005), 1499 – 1507. https://doi.org/10.1016/j.enpol.2004.01.007

[5] Erik HellstrÃĈÂűm, Maria Ivarsson, Jan ÃĈâÃęslund, and Lars Nielsen. 2009. Look-ahead control for heavy trucks to minimize trip time and fuel consumption. *Control Engineering Practice* 17, 2 (2009), 245 – 254. https://doi.org/10.1016/j.conengprac.2008.07.005

[6] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2014. *An Introduction to Statistical Learning: With Applications in R.* Springer Publishing Company, Incorporated.

[7] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. 2014. *Mining of Massive Datasets* (2nd ed.). Cambridge University Press, New York, NY, USA.

[8] Thuy T.T. Nguyen and Bruce G. Wilson. 2010. Fuel consumption estimation for kerbside municipal solid waste (MSW) collection activities. *Waste Management & Research* 28, 4 (2010), 289–297. https://doi.org/10.1177/0734242X09337656

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[10] Federico Perrotta, Tony Parry, and Luis C. Neves. 2017. Application of machine learning for fuel consumption modelling of trucks. In *2017 IEEE International Conference on Big Data, BigData 2017, Boston, MA, USA, December 11-14, 2017.* 3810–3815. https://doi.org/10.1109/BigData.2017.8258382

[11] Chotirat Ann Ratanamahatana, Jessica Lin, Dimitrios Gunopulos, Eamonn J. Keogh, Michail Vlachos, and Gautam Das. 2010. Mining Time Series Data. In *Data Mining and Knowledge Discovery Handbook, 2nd ed.* 1049–1077. https://doi.org/10.1007/978-0-387-09823-4_56

[12] Elnaz Siami-Irdemoosa and Saeid R. Dindarloo. 2015. Prediction of fuel consumption of mining dump trucks: A neural networks approach. *Applied Energy* 151 (2015), 77 – 84. https://doi.org/10.1016/j.apenergy.2015.04.064

[13] Zeng Weiliang, Miwa Tomio, Wakita, and Morikawa Takayuki. 2015. Exploring Trip Fuel Consumption by Machine Learning from GPS and CAN Bus Data. *Journal of the Eastern Asia Society for Transportation Studies* 11 (12 2015), 906–921. https://doi.org/10.11175/easts.11.906

[14] Sandareka Wickramanayake and H. M. N. Dilum Bandara. 2016. Fuel consumption prediction of fleet vehicles using Machine Learning: A comparative study. *2016 Moratuwa Engineering Research Conference (MERCon)* (2016), 90–95.

[15] Min Zhou, Hui Jin, and Wenshuo Wang. 2016. A review of vehicle fuel consumption models to evaluate eco-driving and eco-routing. *Transportation Research Part D: Transport and Environment* 49 (2016), 203 – 218. https://doi.org/10.1016/j.trd.2016.09.008